Restaurant Recommendation Using Sentiment Analysis
CSE158 & 258 Assignment 2

Silver De Guzman
j5deguzm@eng.ucsd.edu
A53212113

Lawrence Shiu
lshiu@ucsd.edu
A13549986

Ravi Sheth
r1sheth@ucsd.edu
A13968500

Abstract

Most times when writing reviews online, people tend to express their thoughts about a specific restaurant using words and phrases that express their experience at that restaurant. Their sentiment, captured by words, is then represented by a star rating between one and five. This report attempts to find an acceptable model that is able to predict the star rating of a restaurant review based on the text of that review.

1. Introduction

As a platform for choosing and getting recommendations on different businesses from coffee places to restaurants to even salons, Yelp is one of the most popular apps for handling those requests. With the app, users have access to all reviews given to a business, the average rating of a business, pictures of products/services offered, amongst other information. What we choose to focus on is the content of the review. Using just the contents of a review, we wanted to see how reviewers would give star ratings based on their sentiment expressed in a review.

Yelp is widely used by a huge variety of people. When looking for recommendations most people tend to read the experiences of other restaurant goers on Yelp reviews. If the reviews are good, then people will be more inclined to go to that restaurant. More often than not, people tend to just look at the overall rating, in star values, of the restaurants, rather than reading all the reviews. We were curious about if there was a relationship between the review text and the star rating that a user would give. If there was a strong relationship then that would suggest that people tend to operate on a similar rating plane, and just looking at the star rating would be a good indicator of whether or not you will like your restaurant experience. If not then, looking at the individual reviews is necessary in order to determine if you would like the restaurant.

The bag of words is the most common approach towards solving this problem. We used many different feature matrices in order to ascertain which will perform with the best results. We used the top 2000 most popular words, the top 2000 most popular adjectives, the tfidf scores of each, as our feature matrices. We used ridge regression in order to make our predictions and then calculated our MSE scores for each matrix.

We finally switched gears and used the latent semantic analysis. We used our tfidf scores of the entire corpus and displayed them in a low dimensional setting. We see lowest recorded MSE values, 0.63, when using latent semantic analysis, as compared to other models.

In this report we will first analyze our findings from the exploratory analysis of our dataset. We will then explain the models we choose to run our dataset on, for example, the bag of words and latent semantic analysis model. We judge the validity of our model based on its MSE scores. We will in the end report our findings for all our models and show that the Latent Semantic model gave us the best scores compared to the rest.

2. Dataset Exploration

The dataset is from Round 12 of the Yelp Academic Dataset Challenge which contains real customer reviews of various businesses in cities across the US as well as outside the US. There are almost 6 million reviews from categories ranging from restaurants to bars to shopping. For the our purposes, we choose to focus on reviews of restaurants located in the US. Table 1 shows some key initial statistics.

Number of Overall Reviews	599696
Number of Businesses	188593
Number of Users	1518169
Number of Businesses (US)	139183
Number of **Restaurant** Businesses (US)	34189
Average Rating Per Restaurant (US)	3.63
Average Reviews Per Restaurant (US)	31.8

Table 1. General Statistics

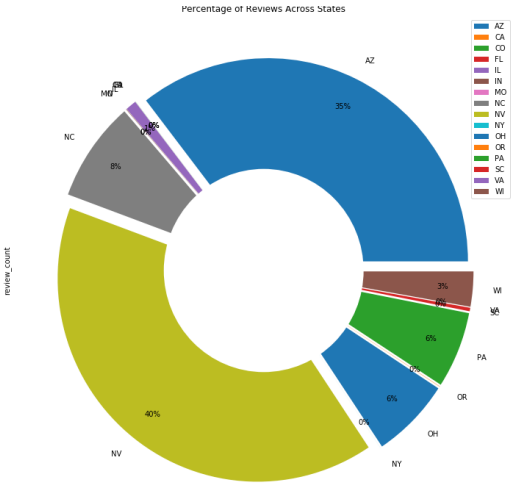

Figure 1. Restaurant Reviews Across US States

2.1. Analysis

From the restaurant reviews of businesses in the US, we find in Figure 1 that most of the data comes from the states Nevada and Arizona. Upon further analysis, we see that the cities Las Vegas and Phoenix have the most text reviews which is what drives the high density of reviews in those states.

The text reviews are in a separate json file from the files containing information related to the business and information related to the user. From the text reviews json file, each line contains the contents listed in Table 2. For our project we only utilize the businessID, userID, star, and text attributes of a datapoint.

Table 2. Review Contents

Feature	Description
businessID	ID for restaurant
userID	ID for customer
star	rating given to business
text	text of the review
useful	# of reviewers that found review useful
funny	# of reviewers that found review funny
cool	# of reviewers that found review cool

2.2. Key Findings

- Since most of the reviews are from the cities Las Vegas and Phoenix, we wondered if people review things differently based on location. We analyzed the top words that appear in reviews for restaurants in Las Vegas and for restaurants in Phoenix. We found that the top words did not differ much and concluded that people do not review differently across different cities. Therefore, we decided to focus our analysis on restaurants reviews in Las Vegas. Note that this finding could be different across countries.

- In the overall reviews for just Las Vegas, there are more 4 and 5 star reviews compared to 1 and 2 star as seen in Figure 2.

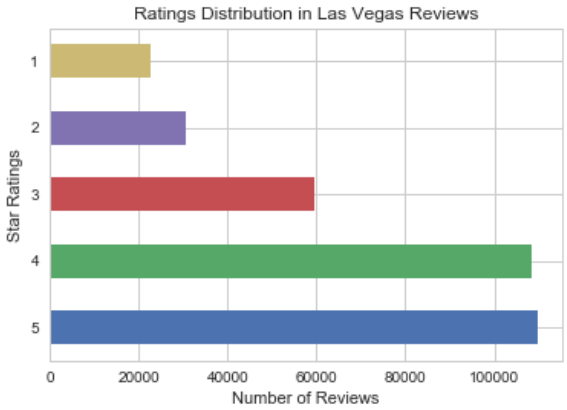

Figure 2. Star Ratings Distribution in Las Vegas Dataset

- From Figure 3 we see that the length of the reviews (i.e. number of words per review) is about equivalent across all star reviews. Furthermore, the number of adjectives per review vs review rating indicates that the number of adjectives in each review decreases slightly as the rating decreases as shown in Figure 4. Since the length of the review does not correlate with the starred reviews, the adjectives are found to be a better indicator of the star rating. Regarding adjectives, the longer reviews contain more adjectives and thus have higher ratings. It is worth noting that this may not be the case in our dataset because there is an unbalanced amount of high and low star reviews (indicated previously).

- From the word cloud in Figure 6, we can see that the most popular words are 'the', 'and', and 'i'. These stop words are found to be common in all the word clouds, (1) for top 1000 words of reviews from a different location, Phoenix, and (2) for reviews from one star and five star reviews. Since there is overlap of the top words in the reviews, irrespective of star rating or location, we decided to look at adjectives to see what type of adjectives people use most frequently in order to describe their overall impression of a restaurant. The most frequent words we see are words describing good sentiment, like 'good' and 'great'. The words that express negative sentiment are also present but at a lower frequency. The likely reason for this phenomenon is that there is a skewed dataset with high rating reviews being more prevalent.

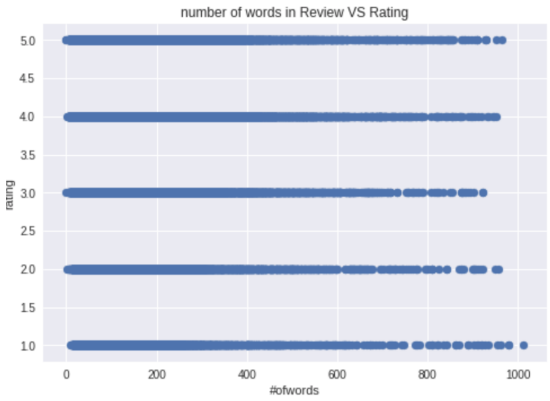

Figure 3. Length of Review vs Rating

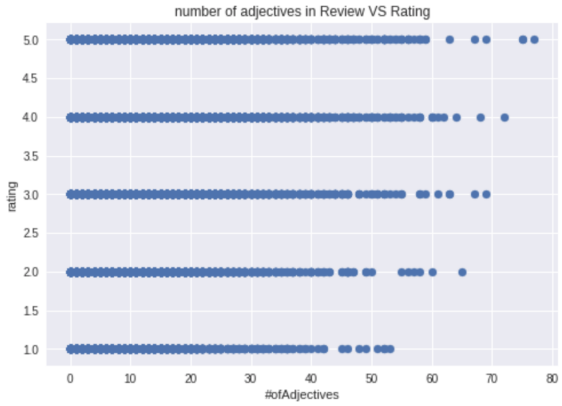

Figure 4. Number of Adjectives in Review vs Rating

- The top 1 star and 5 star reviews with stop words removed still show that there is a big overlap between the words used in both categories. We also see more positive words in the top 1000 most common words. The reason for this, again, is that there is a skewed dataset. Refer to Figure 5.

3. Feature Matrix

3.1. Data Preprocessing

- Removing punctuation from reviews
- Removing capitalization from reviews
- Removing all non-restaurant reviews
- Removing restaurants and users with < 10 reviews
- Using only 100,000 reviews from the most reviewed city (Las Vegas)

From our findings in the previous section, we focus on reviews solely in the city of Las Vegas. There are over 300,000 reviews from Las Vegas alone, but we randomly use a subset of 100,000 reviews.

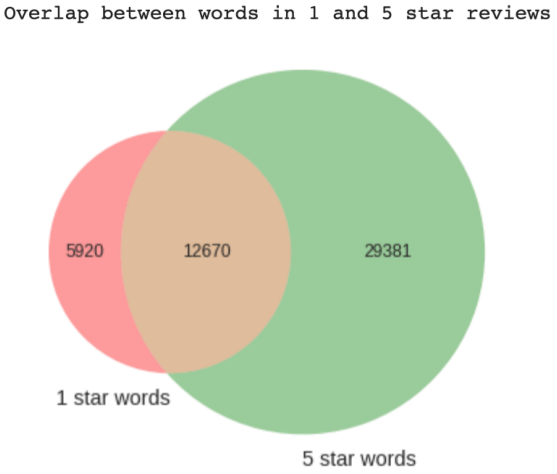

Figure 5. Overlap of words between 1 and 5 star reviews

3.2. Bag of Words

We counted the frequency of each word in the entire corpus. Using the values we get, we built the first feature matrix with the top 2000 most popular unigrams in all the reviews.

Figure 6. Top 1000 Most Popular Words

3.3. Part of Speech Tagging

The next feature matrix we built was using the top 2000 most popular adjectives that are in the corpus. We used the nltk library to classify words in the corpus as adjectives, comparative adjective and superlative adjectives and the rest. We then used the top 2000 adjectives to create the feature matrix.

4. Predictive Task

Given the reviews of (user,restaurant) pairs, we try to accurately predict the rating using only the features of the text. Can semantic analysis findings provide a better indication of rating prediction than other non-text features? We exploit different parts of the text to see what results in the best performance.

Figure 7. Top 1000 Most Popular Adjectives

4.1. Evaluation

A 60-20-20 split on the dataset is done for training-validation-test sets. To evaluate the accuracy of the predicted ratings, we measure the mean-squared error (MSE) values between the predicted ratings and groundtruth ratings:

$$MSE = \frac{1}{N}\sum_{u,i}(r_{ui} - r^*_{ui})^2$$

where r^*_{ui} is the predicted rating that customer *u* gave to restaurant *i*, r_{ui} is the actual groundtruth rating, and *N* is the total number of reviews in the test set.

We decided to use MSE instead of other metrics such as mean absolute error (MAE) since MSE has the property of penalizing large magnitude errors more greatly.

4.2. Ridge Regression

We use a ridge linear regressor to fit our models to our dataset and calculate the θ parameters. An L_2 regularization is included in order to penalize for model complexity. Using θ parameters, we can directly make rating predictions on the validation and test sets.

4.3. Alternating Least Squares

The alternating least squares is similar to gradient descent and works on jointly convex solutions. It minimizes the equation below by iteratively solving for α, β_u, and β_i until convergenc .

$$\arg\min_{\alpha,\beta} \underbrace{\textstyle\sum_{u,i}(\alpha + \beta_u + \beta_i - R_{u,i})^2}_{\text{error}} + \underbrace{\lambda\left[\textstyle\sum_u \beta_u^2 + \sum_i \beta_i^2\right]}_{\text{regularizer}}$$

5. Model

5.1. Baseline Model

For our baseline, we used the average star rating amongst all reviews in our training set and use that as the trivial predictor.

5.2. Model 1: Bag of Words

In our first approach we used the bag of words methodology and used the top 2000 words most popular words as our feature matrix. We then wanted to edit our Bag of Words by removing stop words in our approach. Our thought process was that stop words do not give much information to the semantic of a review, therefore we should take them out and take the top two thousand non stop words.

We then used a tf-idf matrix to offset the high frequency words that appear in all documents and give more weight towards words that appear less commonly across all documents. This will allow the θ values to be even across all words instead of being highly skewed towards less frequent words. Although this may seem the same, by evenly distributing theta values it will decrease our MSE.

5.3. Model 2: Part of Speech Tagging

The second feature matrix we build is using the top two thousand most popular adjectives in the dataset. The assumption was that positive words would appear in reviews with high star ratings and vice versa for negative words. We saw that in our exploratory analysis that the top 50 adjectives showed up a lot more frequently in 4 and 5 rated stars. One reason why we saw those results might be because the dataset is skewed as there are more 4 and 5 rated stars than other starred reviews.

We still thought that this was an avenue worth exploring because our hypothesis was that people would use different positives words for different star ratings. We again also explored the results of a tf-idf matrix to see if there was any improvement by giving less weight towards common words that appear in all reviews.

5.4. Model 3: Latent Semantic Analysis

This model is a very high dimensional tf-idf feature matrix with its entries corresponding to tfidf scores of all words in a review for each review in the training data. LSA works by performing dimensionality reduction to project the high dimensional feature matrix into a lower dimensional representation. In doing so, it keeps the most informative features while discarding redundant features.

While Models 1 and 2 use only the top 2000 words and top 2000 adjectives respectively, this model uses all the words in the corpus excluding stop words. This is fine to do since the resulting feature matrix will be projected to a lower dimensional space in the end. The number of dimensions to keep is part of the hyper parameter tuning.

A row in the feature matrix corresponds to the TF-IDF vector of scores for a single review. This becomes a $12,000 \times 71,030$ matrix. After applying PCA, the columns dimension is greatly reduced.

5.5. Model 4: Customer-Business Biases

While we mainly seek to garner how text features can predict the rating of a restaurant review, we also decided to experiment with a model that incorporates non-text features. This model instead utilizes a customer's and a restaurant's individual biases to see how that influences the rating. Does a model that uses past history of ratings a customer gave and of ratings of a business do better than sentiment analysis models? If a customer tends to give high (or low) ratings based off previous history, then that customer is more likely to give a high rating to a restaurant they have not yet visited.

A similar pattern applies to a restaurant. If a restaurant has on average more 4-star and 5-star ratings than 1-star and 2-star ratings, then that is indicative of future ratings. This can also influence a customer's rating in the sense that if previous customers gave a high rating, then maybe this restaurant deserves a high rating since the popular consensus seems to agree so.

The customer-business biases model is an equation of the form:

$$Rating(u, i) = \alpha + \beta_u + \beta_i \quad (1)$$

where α is the mean of the ratings, β_u is a vector of customer biases, and β_i is a vector of restaurant biases. We fit the training data into the above rating equation and run the method of alternating least squares to find the best approximation of the α and biases. Once these values are determined, the rating of a particular customer-business pair is a matter of solving the equation.

6. Literature

This yelp dataset that includes text, rating, location, businesses, users, etc has had many studies on it. There even is a Kaggle yelp challenge currently available, which is where we pulled our dataset from. Studies that have been performed on this dataset have ranged from location extraction, social graph mining, or even something like restaurant closures.

Our predictive task was to predict the star rating given the text of the review. For the most part we have used Bag of Words and variations of it. This technique of semantic analysis came originally from Evgeniy Gabrilovich and Shaul Markovitch as a means of improving text categorization and is called Explicit Semantic Analysis (ESA). The difference between ESA and our process in this project was that ESA also uses a cosine similarity to see the relatedness of two words.

Another established technique, is the Latent Semantic Indexing (LSI) created by Scott Deerwester and Susan Dumais. This is a technique that uses natural language processing and distributional semantics. The concept of this technique is that similar words will be in similar places of documents. Then it uses singular value decomposition (SVD) to reduce the amount of rows while retaining the structure of the data and once again use cosine similarity to detect how similar two words are by their vectors.

7. Results

7.1. Bag of Words

Bag of Words used the top 2,000 most common words and used a regressor to give each of those words a theta value. This improved our MSE to 0.74 from our baseline solution of 1.44.

We then tried taking out all stop words, but contradictory to our thoughts, removing stop words actually increased our MSE value. We later found that lower starred reviews tend to have higher word length (table 3) and that the stop words could be representing the word length of a review. I.e more stop words would mean higher review length. By taking out stop words we would be removing a useful classifier and therefore increase our MSE.

Stars:	1 star	2 star	3 star	4 star	5 star
Average Length:	809	869	832	785	654

Table 3. Average review length

Since removing stop words did not improve our MSE value, our next thought process was to use tf-idf score to offset high frequency words and give more weight towards less common but more impactful words. By using tf-idf scores we lowered our MSE by .04 from 0.74 to 0.70. The logic behind this is that we are able to keep stop words, but to give stop words or all frequent words less weight, and to give less frequent words more weight when it comes down to deciding sentiment.

Some other things we found from Bag of Words was that negative words have more weight than positive words. This can be explained such that if a negative word appears, there will be a low probability that a high review was given. On the other hand if a positive word appears, the review can still be negative since the reviewer can express positivity in some parts, but an overall negative impression. This results in negative words being a better indicator of the review given.

Top Words	Weight
Amazing	0.2425
Outstanding	0.2421
Excellent	0.2316
Awesome	0.2307
Perfection	0.1837

Top Words	Weight
Worst	-0.6686
Horrible	-0.6378
Rude	-0.5217
Terrible	-0.4310
Mediocre	-0.3984

Figure 8. Top 100 Positive Words

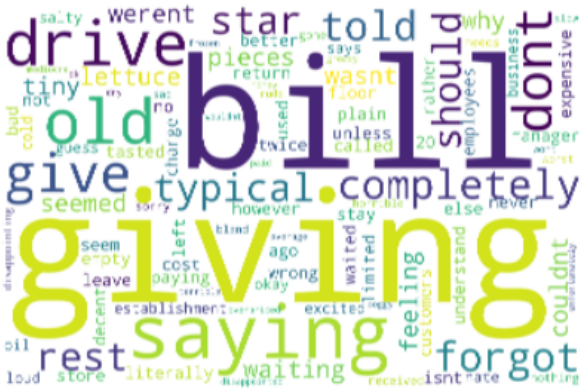

Figure 9. Top 100 Negative Words

7.2. Part of Speech Tagging

We used part of speech tagging to get all the adjectives in the corpus. We believed that using words that expressed sentiment, would allow us to more accurately classify the star rating of a review. Meaning we would primarily see words that express strong liking like, "excellent" or "amazing" in five star reviews, while words like "abhorrent" or "disgusting" would be more prevalant in 1 star reviews. We believed we would see that pattern for reviews in the 2,3,4 star category as well, where some words are either dominantly or exclusively seen in one of the star categories. We then used a regressor and got an MSE value of 1.033 as seen in Figure 10.

We believed that the reason for this error could be that our feature matrix was getting incorrect answers due to the fact that all the adjectives in the corpus were being used. We thought using the more popular adjectives (the top 2000) would help increase our accuracy. Reducing the feature matrix size did help a little bit but not still not good enough. We recorded a 1.0329 MSE value as seen in Figure 10.

We believe we might be getting poor results because of the skewed dateset. There are a lot more highly rated restaurants in the dataset than lower rated restaurants.

MSE values using different feature matrices

Feature Matrix	Top 2000 most popular words	Top 2000 most popular words without stop words	All adjectives	Top 2000 adjectives
MSE	0.7474	0.7575	1.0330	1.0329

Figure 10. MSE Values for different features

7.3. Latent Semantic Analysis

The performance was greatly affected by fine tuning the number of dimensions K that were kept. It was also found that removal of stop words helped the MSE decrease from 0.65 to 0.63. Clipping the predicted ratings to be in the range 1 to 5 improved performance by 8%.

Figure shows 11 shows how the MSE varies with changing λ and the latent factor components K. The best λ was found to be 0.01 but between $\lambda = 0.1$ and $\lambda = 0.01$, performance was very similar with differences of a thousandth of a point. The value of K that gave the best MSE was 500. The best MSE was 0.63 corresponding to $K = 50$ and $\lambda = 0.01$.

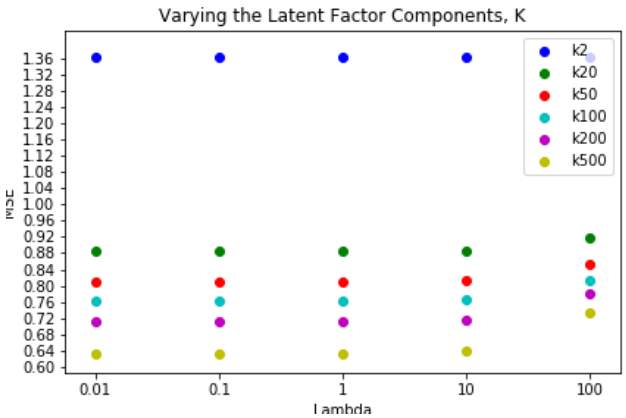

Figure 11. Fine tuning latent factor parameter in LSA model

7.4. Customer-Business Biases

We did not exploit this model much since it was merely used for comparison with using textual features. This is a very simple model and because of it's simplicity it did not perform well when compared to a model that used reviews of a text. Still, it does better than the baseline.

7.5. Performance Comparison of Models

Table 4 shows the best MSE values that were found for each of our models with slight variances of the feature vectors in the model.

Model 1 was the Bag of Words model using the top unigrams. We vary Model 1's feature vectors in the following

way:

- word count of top 2000 unigrams, keep stop words
- tf-idf of top 2000 unigrams, keep stop words
- word of top 2000 unigrams, remove stop words

Model 2 was the Part of Speech tagging model using the top adjectives. An adjective by definition does not include stop words. We vary Model 2's feature vectors in the following way:

- word count of top 2000 adjectives
- tf-idf of top 2000 adjectives

Model 3 was the Latent Semantic Analysis Model. It uses the tf-idf feature vectors of entire corpus of words in the training set. We experiment with and with out stop words for $K = 500$ dimensions:

- tf-idf, keep stop words, K=500
- tf-idf, remove stop words, K=500

Model	Best MSE
Baseline: Average Rating	1.44
Model 1: BOW (Word Count)	0.74
Model 1: BOW (TF-IDF)	0.70
Model 1: Removing Stop Words	0.78
Model 2: Adjectives (Word Count)	1.10
Model 2: Adjectives (TF-IDF)	1.08
Model 3: LSA (remove stop words)	0.65
Model 3: LSA (keep stop words)	0.63
Model 4: User Preference	1.19

Table 4. MSE of Different Models

8. Conclusion

We tried many models and variations of those models to find the MSE, but the two best models were Bag of Words using TF-IDF and Latent Semantic Analysis. Our baseline was using the average rating as the prediction with a MSE of 1.44 and BOW gave us a MSE of 0.74 while LSA gave us a MSE of 0.63 which halves our baseline MSE.

It was interesting to see how using only adjectives or removing stop words had worse performance than merely using all top 2,000 words. As shown in Figure 6, many stop words in the top common words. That means that stop words such as "the", "at", "a", etc actually carry some weight when determining the sentiment of a review. One likely answer to this is that length is also a determining factor in predicting star rating and that stop words indirectly correlate with the length of the review.

While Bag of Words by itself provided accurate results, using Bag of Words with TF-IDF scores improved our MSE further by 4%. Although Bag of Words does a good job of determining sentiment, BOW gave certain words higher weights and skewed our feature matrix greatly. By using TF-IDF score, we were able to make our theta values more uniform and therefore, give us better predictions.

LSA proved to be the best model overall because it is able to learn patterns and relationships of a word and review's lower dimensional feature representations. As mentioned before, fine tuning the number of latent dimensions to keep provided a drastic improvement on MSE. This makes sense. If too many dimensions are discarded, we may lose much of the dimensions that carry greater variance. We want to preserve that variance but discard the low variance and redundant dimensions. Discarding too little dimensions also affects performance.

9. References

1. Yelp, Inc. Yelp Dataset. RSNA Pneumonia Detection Challenge — Kaggle, 2 Aug. 2018, www.kaggle.com/yelp-dataset/yelp-dataset.
2. Latent Semantic Analysis. Wikipedia, Wikimedia Foundation, 18 Nov. 2018, en.wikipedia.org/wiki/Latent_semantic_analysis.
3. Latent Semantic Analysis. Wikipedia, Wikimedia Foundation, 18 Nov. 2018, en.wikipedia.org/wiki/Latent_semantic_analysis.